***Ambiguous Requirements***

***SUBMITTED BY:***

***Haseeb Azam Sheikh***

***(****131374****)***

***SUBMITTED TO:***

***Sir Ahmad Mohsin***

**Ambiguity**

**Ambiguity** is when the meaning of a word, phrase, or sentence is uncertain. There could be more than one meaning.

Typically, it is best to avoid **ambiguity** in your writing. When you make statements that are **ambiguous,** you confuse the reader and hinder the meaning of the text. However, sometimes **ambiguity** is used deliberately to add humor to a text.

**Examples of Ambiguity in Literature and Speech**

In "The Rose" by William Blake, there is enough ambiguity about what is meant by "rose," "sick," "bed of joy," and "worm," that the poem is open to many interpretations:

"O Rose thou art sick.

The invisible worm,

That flies in the night

In the howling storm:

Has found out thy bed

Of crimson joy;

**AMBIGIOUS REQUIREMENTS**

Software Project managers have realized that ambiguity in software requirements can create greater harm than ambiguity or defects in any other stage of software development. Yet, unfortunately, most software projects, still fail to understand the importance of validating requirement specifications, thanks to the pressure of releasing products faster to the market.

**Lexical Ambiguity**

[Lexical ambiguity](http://en.wikipedia.org/wiki/Polysemy) occurs when a word has several possible meanings, resulting in a sentence having multiple possible interpretations. In the sentence:

*I like writing.*

## Syntactic Ambiguity

[Syntactic ambiguity](http://en.wikipedia.org/wiki/Syntactic_ambiguity) occurs when a sequence of words can be given more than one grammatical structure. For example, consider the sentence:

*The police shot the rioters with guns.*

**Examples**

**Negative requirements**  
 Negative, or inverse, requirements state what the system will not do. Here's an example from an actual project: "All users with three or more accounts should not be migrated." Try to rephrase negative requirements into a positive sense: "The system shall migrate only users having fewer than three accounts." When changing a negative requirement into a positive one, you often need to insert the word "only" to clarify the conditions that permit the system action to take place.

**Adverbs**  
 Adverbs are subjective and qualitative, and they inevitably result in diverse individual interpretations. Here's an illustration from an actual specification: "Generally incurs a 'per unit' cost…" But this requirement did not provide any indication regarding the conditions under which we would not incur a per-unit cost or what to do then.

**Conclusion:**

If we want to decrease the ambiguous requirements we should write complete requirements and also use precise language that communicates information across domain to reader. We need to monitor the effectiveness of our communication, and balance that with the amount of time.

In first lexical ambiguity shows incomplete information like I like writing but writing what?

In second syntactic ambiguity is about grammatical mistake that may confuse the other to understand it in some ways.

**References:**

* <https://shane.io/2013/04/22/ambiguity-in-software-specs.html>
* <https://link.springer.com/chapter/10.1007/978-1-4615-0465-8_2>
* <http://www.cigniti.com/blog/how-harmful-can-be-ambiguous-software-requirements/>